A Computer Vision and Hearing Based User Interface for a Computer Game for Children

Perttu Hämäläinen[1] and Johanna Höysniemi[2]

[1] Helsinki University of Technology, Laboratory of Telecommunications Software and Multimedia, P.O.Box 5400, FIN-02015 HUT
perttu.hamalainen@hut.fi
[2] University of Tampere, Tampere Unit for Computer-Human Interaction,
Kanslerinrinne 1, FIN-33014 University of Tampere
johanna.hoysniemi@uiah.fi

Abstract. This paper describes the design of a perceptual user interface for controlling a flying cartoon-animated dragon in QuiQui's Giant Bounce, a physically and vocally interactive computer game for 4 to 9 years old children. The dragon mimics the user's movements and breathes fire when the user shouts. The game works on a PC computer equipped with practically any low-cost microphone and webcam. It is targeted for uncontrolled real-life environments such as homes and schools.

1 Introduction

Computer vision and hearing[1] technologies have been used in various human-computer interaction applications. However, there is little written about designing computer vision and hearing-based computer games and user interfaces for child users in uncontrolled real-life conditions, for example homes and daycare centers. The main difference to installations and exhibitions, such as the Kidsroom [17], are that the lighting and the position of the cameras cannot be assumed to be constant. The technically illiterate end users must also be able to setup the software. Ideally, initialization and adjusting of technical parameters should be as automatic as possible. The games developed by Intel and Mattel [4] come close to that, but they still have usability defects, like voice prompts that instruct the user to step aside so that the computer vision can be properly initialized.

Perceptual user interfaces are interesting when designing computer games, since they enable the user to naturally act out the role of the main character. The natural user interface can deepen the user's emotional commitment to the game and make the game more compelling and exciting. Properly designed perceptual user interfaces can also add physical exercise to the game. This is a benefit compared to traditional

[1] We use the term computer hearing for consistency with the term computer vision, emphasizing that it refers to an electronic sense, although terms like machine listening and computer audition are more common in the literature.

computer games, the overuse of which may have negative effects on children's physical health (see e.g. Subrahmanyam et al. [22]).

This paper describes the design of the first game task and user interface of QuiQui's Giant Bounce, a computer game controlled with movement and voice and targeted at 4 to 9 year old children. The novelty of the approach is that the technology is designed for children in their natural environment from the point of view of usability. The user interface also combines immediate response to both motion and voice input to deepen the illusion of being the main character of the game.

The user interface uses a webcam and a microphone to see and hear the user. The main character and avatar QuiQui mimics the user's movements and lets out a fiery breath when the user shouts, as shown in Figs. 1 and 2. The game is a work in progress and it is designed to have several game tasks. Each task has its own way of moving; for example swimming, flying or jumping. In the game task presented in this paper, QuiQui has to fly through the clouds over a desert to make them rain.

The paper first discusses the technical requirements of the user interface from a usability point of view. Then a set of simplifying assumptions is defined, aiming at a practical implementation that satisfies the requirements. Finally, the design and implementation of the user interface is presented together with the findings of the testing with child users.

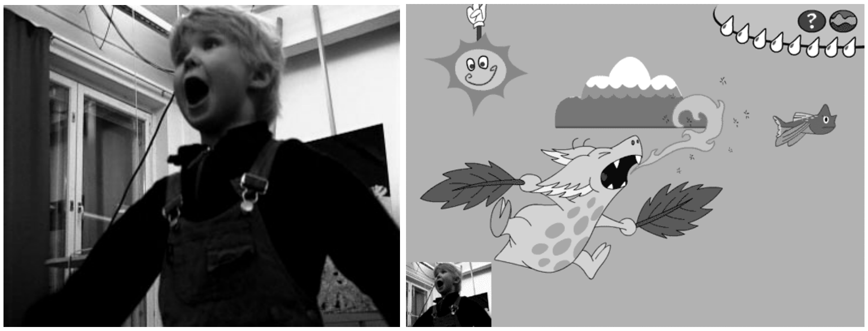

Fig. 1. QuiQui breathes fire when the user shouts

2 Background

Computer vision and hearing were chosen as the interaction technology for QuiQui because we wanted to provide a natural and unencumbered user interface that would also be robust and accessible in every home. We did not want to use any sensors attached to the user's body. Such sensors are easy to break and are also awkward if several children want to play the game together or by taking turns. Because of the variety of webcams on the market, we did not want to optimize the game for a particular hardware setup. Although this adds a number of unknown variables such as frame rate, optical parameters and noise, game requires no purchasing of specific hardware. Thus, it can be simply downloaded from the project's homepage, *http://www.kukakumma.net.*

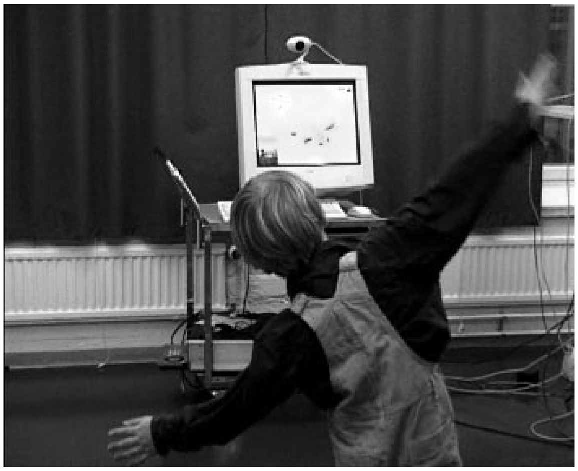

Fig. 2. A typical setup of the game: The camera is mounted on the monitor and the user moves in front of the screen

Much work has been done in the field of perceptual user interfaces based on computer vision and hearing. Our work is closely related, for example, to the Perceptive Spaces and physically interactive story environments developed at MIT Media lab [23][17]. These projects include several applications where one or multiple human users are tracked and the sensory data is used to control a graphical representation of the user or the reactions of the other characters.

Considering game applications, QuiQui's peers can be divided into two categories based on the user's representation in the game. In the first category are games that use a "video avatar", that is, the video image of the user or a part of it is directly interacting with computer graphics, as shown in Fig. 3. The approach was experimented as early as 1984 in Myron Krueger's pioneering work Video Place [15] and it has been commercialized by Intel and Mattel [4], as well as by Vivid Group [24].

QuiQui's Giant Bounce is an example of the second category, that uses a computer-generated avatar in the same way as traditional computer games. The avatar mimics the user's movements. In general terms we are talking about human motion capture, which has been researched intensively, as seen for example in Moeslund's survey of more than 130 related papers [16]. However, the solutions presented in the literature are rarely applicable to computer games in home and school environments. One reason for this are the requirements of the users and the target environment, elaborated in the next chapter. Also, in consumer applications the methods used can never be fast enough because the more effective the algorithm, the wider range of computers can run the program.

Games often restrict the space of possible movements, which is fortunate from the point of view of technology. Exploiting the knowledge about possible movements may enable solutions that would not work in other contexts. Following this principle, several user interfaces for existing games were developed by Freeman et al. at Mitsubishi labs [7][8][9]. However, designing a user interface for an existing game is restricted compared to designing the game and user interface together. For example, in games in which the avatar jumps when the user presses a button, the height of the jump is usually constant. A perceptual user interface should allow the user to jump to different heights, which requires that more control data is extracted from the user's movements.

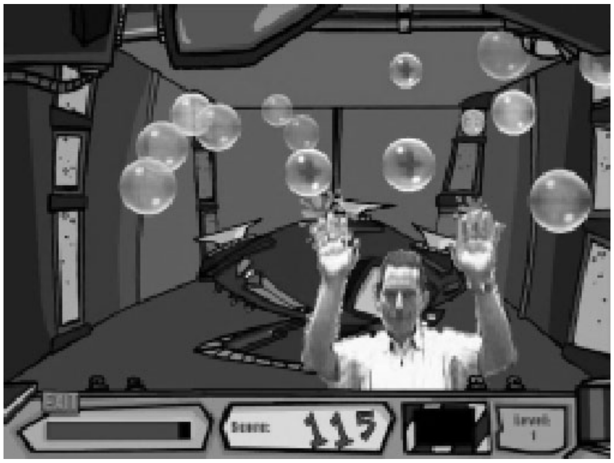

Fig. 3. An example of the "video avatar" games [4]

3 Requirements and Assumptions

In this paper we have mostly developed further the ideas presented by Freeman et al. [7][8][9]. Considering the target group and the environment of use, the challenge was to find a technical solution that does not sacrifice usability and approachability. For this we defined the following requirements:

1. Completely automatic operation without any learning stages, initialization or settings that need user participation. Our goal is that children can use the game without adult guidance.

2. The methods must adapt rapidly to changes in the environment, including lighting and camera position.
3. The methods must adapt to the differences in various camera models, including frame rates in the range 15...30 fps, noise, motion blur and color resolution.
4. The system responds with as low latency as possible.
5. The system must tolerate several visible humans, either one player and viewers or several users participating collaboratively.
6. The user must not have to wear any specific clothing or markers.
7. As low computational complexity as possible to enable maximal computational resources for the actual game application.

The requirements are based on the three requirements for human-centered perceptual user interfaces by Crowley et al. [3]: robustness and autonomy, low latency and privacy protection, the last of which is irrelevant here since it applies mainly to multi-user applications and the video footage of the user is not stored in QuiQui's Giant Bounce. As Crowley et al. put it, *"Usability determines the requirements for technological innovation."*

The requirements 1 and 2 rule out many of the existing computer vision systems, for example all the systems based on an assumption of a constant or almost constant background. This includes the technology behind Intel and Mattel's games [4], most of the MIT Media lab experiments mentioned and also the only webcam based game study we know of [19]. Assuming a constant background is lucrative for computer games, because a two-dimensional silhouette of the user can be obtained simply by subtracting every video frame from a sample of the background. Once a silhouette is obtained, simple shape recognition techniques can be applied to produce the control signals needed for a variety of games, as shown by Freeman et al. [7][8][9].

The requirements 1 and 2 are based on the fact that our target group is illiterate and technically incompetent, which makes it difficult for the software to instruct the user. Unexpected changes in the environment do happen when the system is used at homes and the users are children, as D'Hooge points out [4]. Our own usability tests also verify this. Intel and Mattel use voice prompts to guide the user for example to step away from the camera view for initialization [4], but we find that awkward.

The requirement 3 has severe implications on the computer vision methods. The low color resolution, noise, and unpredictable, often automatically controlled color settings such as white balance imply that color based skin and face detectors (see for example Bradski [2]), cannot be used reliably. Fitting curves to image contours (see for example Blake and Isard [1]) is unreliable because of noise and motion blur. The low frame rates decrease the accuracy of predictive methods. The motion of the user may have sudden changes and because of the frame rate, the user may suddenly be hundreds of pixels away from the predicted location. For example in a platform jumping game, the user may stay still, waiting for the right timing, and then suddenly bounce in any direction. An example of the amount of motion blur and changes between successive frames is shown in Fig. 4. The figure contains two frames of a jumping user recorded with a Logitech Quickcam Express webcam in daytime home conditions.

The requirement 4 rules out various gesture recognition methods that recognize movements after they have been completed. Low latency is important for the overall feeling of the game. For example, in a game played by jumping, latency can cause the virtual world to feel sticky, like the avatar was standing in a pool of thick mud.

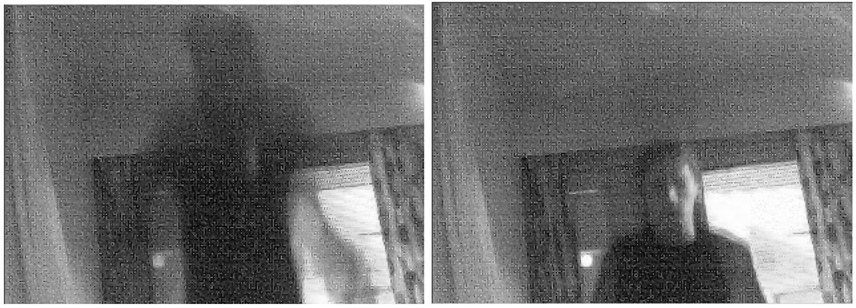

Fig. 4. Two successive frames captured by a webcam when the user is jumping

The requirement 5 is based on our usability tests and the findings of Inkpen et al. [14] and Stewart et al. [21]. It appears that gaming is a social event for the children. It is hard to restrict the situation to one visible user since the others want to watch, standing or sitting beside the main player or for example crouching on the floor. In our tests, children also liked playing the flying game together with a friend, so collaborative operation should be supported when possible.

Requirement 6 further restricts the computer vision methods. It is defined to enable more convenient collaborative play and the delivery of the game in fully digital form as a simple downloadable software package.

Adding the requirement 7 to all the previous ones, one may wonder what is possible in practice. Fortunately, game applications allow many contextual simplifications. We use the following assumptions:

1. The background is temporally "piecewise" constant. If there are changes, they are either immediate or slow and gradual. This includes changes in lighting, sudden movements of the camera (if a child for example bumps into the table on which the camera is placed), and changes in the physical environment, like closing or opening of a door. According to our experience, it also includes automatic adjustments of exposure and gain done by the webcam drivers, since they usually happen slowly or in discrete steps.
2. The user's movements are restricted. The user faces the camera because the camera is mounted near the screen. Distance from the camera is limited because the user must be close enough to the screen to see the graphics and also because there is a limited amount of space available at people's homes. Because of the spatial constraints the user's legs do not necessarily fit into the camera view and we restrict the tracking to the upper body of a standing user, a convention also used by Intel and Mattel [4]. The movements are also often restricted to a set

specific to the game context, a principle often exploited by Freeman et. al. [7][8][9].

3. If there are more than one people visible, the user is not severely occluded by others. This is natural since the user usually wants a clear line of sight to the computer and thus to the camera.
4. Responsiveness is more important than correct tracking of the user. It doesn't matter if the game can be "cheated" using movements other than intended, as long as controlling the avatar is most convenient using the intended movements so that the "right" way of moving is encouraged.

In our opinion, the assumption 4 is the key to designing practical computer vision and hearing methods for perceptually interactive games. In our tests with children, we have noted that the game experience is generally not degraded if the player can cheat the game. Many times it only challenges the child's creativity. For example when testing an early prototype of the flying game, a six year old girl started experimenting with different movements. In the interview after the testing she stated : "The best way to get up in the air is to first flap your hands and then you can jump up and down. " She was clearly satisfied about her finding. The negative side is that the feedback given by the game can be confusing when the user is learning to play the game.

4 Design and Testing of the User Interface

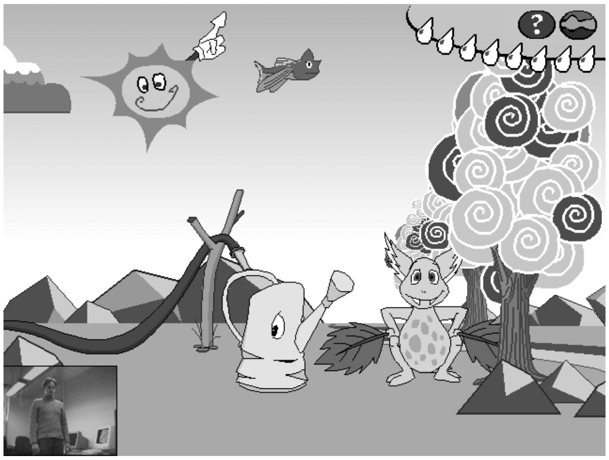

Fig. 5. Starting the flying game

This paper presents the user interface of the first game task of QuiQui's Giant Bounce, a game of flying. The treatment is focused on the *human-avatar interface*, that is, the mapping of user's actions to avatar's actions and the technology behind this. The overall game design and the *avatar-virtual interface* are beyond the scope of this paper. By avatar-virtual interface we mean the interaction happening in the game world, the design of which is more close to user interface and usability design in general, including metaphors, logic, information design etc.

In the context of computer games, the definition of usability is rather vague (see for example the discussion in ACM CHI-WEB archives [10]). The basic contradiction is that a good user interface tries to make things easier for the user, but a computer game needs to present a challenge to the user. Our view is that the human-avatar interface should be as intuitive and transparent as possible. It should let the user focus on the challenge presented by the avatar-virtual interface, to let the user think and act from the point of view of the avatar. It is a usability issue, if the user, as his or her real-world self, does not know how to move so that the avatar flies to a certain direction. However, it is a more general game design problem to for example place the clouds in the sky so that the user, as the avatar, can find them in a certain time.

Note that the division into the human-avatar and avatar-virtual interfaces is not clear when considering game physics. Game physics are the general laws controlling the avatar's movement in the virtual world. The game physics largely define the overall feeling of the user interface, described with adjectives like light, heavy, fast, slow etc. Variables such as gravity have an effect on how the avatar reacts to the user's movements. However, they also have an effect on how the avatar interacts with the game world. Properly designed game physics should increase the feeling of being in control of the game. On the other hand, in games like Liero [26], the game physics constitute the main challenge and enjoyment of the game. The physics are adjusted so that the game requires lightning fast reflexes and decisions when the avatar is flying through the air or swinging at the end of a rope.

In general, it seems that the human-avatar interface is designed to be as simple as possible, with the exception of games like Tomb Raider [27] or Die By The Sword [25]. Those games offer increased realism with a more complex and expressive interface that enables actions such as somersaults or separate control of the avatar's upper and lower bodies.

QuiQui's Giant Bounce uses simple sets of movements that vary according to the game task (sub-game). The user interface design is based on the game design and narrative. In the flying game presented in this paper, the basic setting we started with was that QuiQui has to help a yellow watering can to water the dry desert. This is done by flying through the clouds on the sky to make them rain. QuiQui holds a pair of big leaves in its hands so that they act like wings. The game starts with a metaphorical instruction spoken by the watering can "Flap your hands like a bird to fly up to the sky". A screenshot of this is shown in Fig. 5.

4.1 Methodology: Usability Testing With Functional Prototypes

For example Druin [5] and Hanna et al. [11] have written about collaborative design and usability testing with children. Druin et al. present a method called Cooperative Inquiry for incorporating children in the design process, for example through collaborative low-tech prototyping sessions. Hanna et al. give guidelines for adapting conventional usability testing methodology for children.

Since we were interested in aspects such as the intuitiveness of the user interface, we adopted an approach of iterative usability testing and interviews with users that are new to the game. Druin suggests a long-term relationship with a group of children [5], but in our case such approach would be more useful considering the overall design of the game or for example the story and character design.

The main testing method used was peer tutoring [12]. The test was started with the metaphorical instructions and with instructions to fly through all the clouds. After that the test subject tried to learn to fly and instructions were given only if they were not successful or seemed frustrated. After the test subject had played the game for a couple of times, it was his or her turn to teach the next child to play. We used two approaches, where one child tutor instructed one child tutee and where two child tutors instructed one child tutee. We used the peer tutoring method because we wanted that children are able to teach other children how to use the game, which is important in social settings such as schools.

The tests were conducted at homes and at a preschool and school. The tests were videotaped to record the movements and gestures of the users. The tests were done using perceptually interactive prototypes, initially with more simplifications than in the final version. Although there has lately been some research on prototyping tools for perceptual user interfaces [18], they are more suitable for user interfaces where the user for example points at things. In skill-and-action games such as the flying game, the interaction is more fast-paced and prototyping is more difficult.

Two iterations of the user interface were produced based on tests with 12 and 16 children, respectively. The subjects were of ages 5 to 9. In addition to the second test, the final prototype has been tested informally with several users. Informal tests were also used between the two usability tests to try out various technical implementations of the second and final user interface. We have also had feedback from teachers and parents who have downloaded and installed the game.

4.2 Results

We evaluated the different iterations and technical implementations of the user interface based on how quickly the users were able to fly through all the clouds. Only the human-avatar interface was changed between the tests. Other aspects of the game, such as the locations of the clouds, were not modified. The results of the second test were that all the children were able to play the game and the average playing time dropped by 66% from the first test. All children learned to fly and could fly through all the clouds. The children also wanted to play the game several times, which indicates that they liked the game. In the informal tests it was also found out that the game is suitable for at least some 4-year-old children.

4.2.1 Interaction Model

In addition to the conclusive results the tests provided significant information during the design process. In the first test, the most frequent reaction to the metaphorical instructions was that the children waved their both hands to fly and bent their upper bodies sideways to control the direction of the flying, as shown in Fig. 2. In the first prototype our initial suggestion was that the user would use only one arm to fly sideways. This was changed to match the observations in order to make the user interface more intuitive.

A screenshot of the second and final prototype is in Fig. 6, showing the user flying to the left.

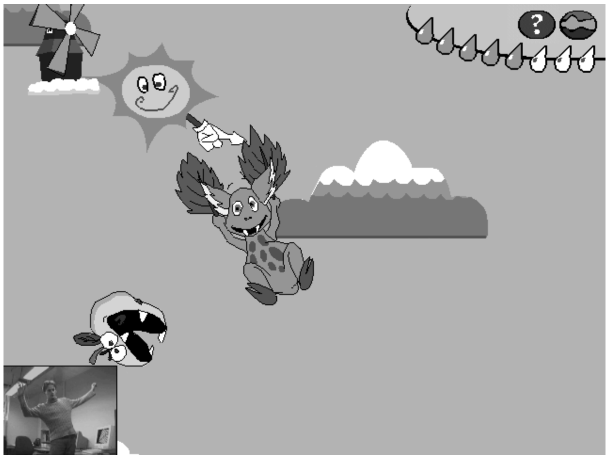

Fig. 6. The flying game in operation

4.2.2 Technical Design

From the point of view of technical design, the main point learned was that the variety and flexibility of children's movements are very difficult to anticipate, even if certain ways of moving were more frequent than others. This directed the computer vision methods towards analyzing holistic and rather vague image features such as image moments instead of precise tracking of the user. Examples of the various ways the children moved are shown in Fig. 7.

4.2.3 Spatial Design

We also encountered problems with the spatial design of the user interface. Because of the limited space available at schools and especially at homes, the camera cannot be placed so that the user is always visible. Instead, the user must be guided to stay within the field of view of the camera. This is problematic especially when an exact one-to-one mapping between the user and the avatar is not possible. For example when the avatar is flying in the air, there is no intuitive mapping for the user moving sideways. Because of this, there is a gap in the feedback of the user interface, and the user can accidentally step out of the camera view. This happened several times in the usability tests.

The problem was initially tackled with the small camera view window in the bottom left corner of the screen, as shown in Figs. 5 and 6. It helps in verifying that the user is positioned correctly, but it was found out that it was hard for the test users to monitor their position. It appears that the camera view is mainly helpful for adults in determining the correct placement of the camera. In general, the younger the users, the easier they forgot that they should stay near a certain optimal location.

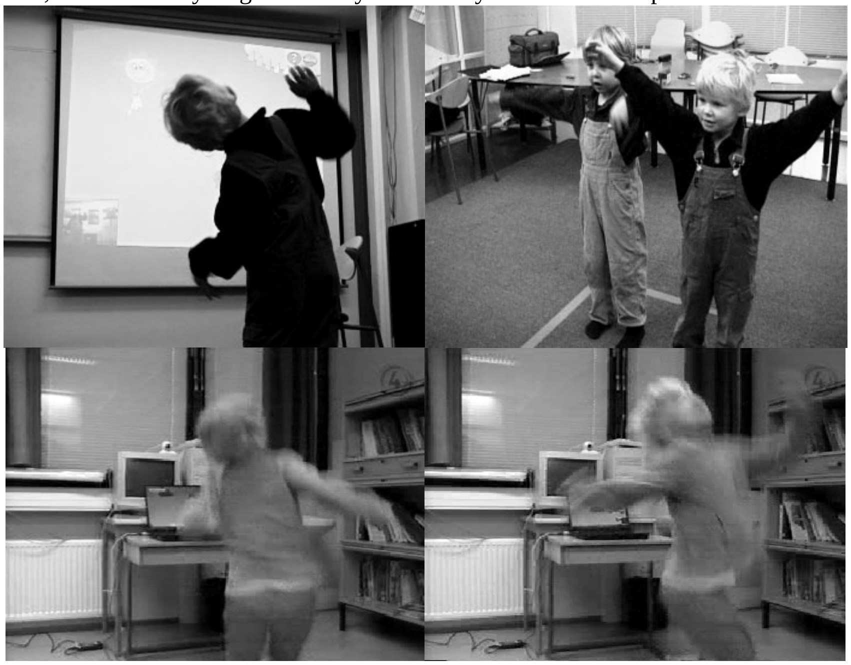

Fig. 7. Test users learning to fly. Note that the figures at the bottom are blurred because of the rapid movements of the user

We did not want to abandon the concept of a flying avatar. Also, we did not want to set more technical constraints for the game, such as a specific camera model with a wide enough field of view. Our solution was to create an approximately one meter wide "magic square" on the floor using marker tape. Our tests show that this considerably helps the children to remember the spatial constraints. They seemed to understand it easily when we explained that "You must stand in the magic square so that the eye of the computer (the camera) can see you." This instruction was usually the first one given when the children explained each other how to play the game. Although adults might regard things like magic squares as cheap compromises or cheating, magic is a useful concept when designing children's technology. It is important and natural in children's fantasy and play. Another good example of using magic to overcome difficult design constraints is given by Druin [6].

The difficulties encountered were caused by the real world constraints, i.e. limited playspace and field of view of the camera, and the mismatch between the game and real world physics. The magic square is not necessarily the right solution for all games. For example, in a platform jumping game the game physics may match the physical world, but moving in the virtual space becomes a problem if the virtual space is larger than the real playing space. One solution is to map the location of the user to avatar velocity, that is, to map physical variables to derivatives of their virtual counterparts. An example of this is the SURVIVE system [23].

4.2.4 Cognitive Load and Feedback to the User

We also encountered problems that were apparently caused by the high cognitive load of the user interface. The main observation was that it is hard to get the user's attention when he or she is learning to fly. The user is also further away from the screen than in traditional computer games, so that the graphics must be scaled, making the avatar larger in relation to the size of the screen. This further complicates things as the user has less time to react when flying around. Although we tried to keep the visual appearance of the game as simple as possible, there was so much happening that the test users did not notice everything. For example, the user's did not always notice the raindrop symbols of the top right corner of the screen. New colored raindrops are added when QuiQui flies through a cloud.

We did not yet find any general solutions to provide feedback that inevitably catches the user's attention. It seems that rather extreme techniques might be needed, such as cinematographic close-ups of important events.

5 Technical Design and Implementation

This chapter describes and evaluates the technical design and implementation, including game physics, computer hearing and computer vision. The design is an example of a simple and practical technical solution that relies on the requirements and assumptions presented in chapter 3.

The general design principle is to use as low-level features of the image and sound data as possible, considering the game context. An example of a low-level image feature is the center of mass, in contrast to a high-level interpretation such as three-dimensional pose of the user.

The more decisions and interpretations are made, the more there are chances for errors. Since the environment of use and the actions of the child users are unpredictable, it is difficult to ensure that the assumptions behind the decisions and interpretations hold in every situation. A simple motion tracker that interprets various movements as flying is better than a sophisticated tracker that causes the avatar to fall down when it makes a false interpretation and moves the avatar's legs instead of its arms.

The main benefits of the approach are robustness and simplicity. The main drawback is the dependency on the game context, that is, the loss of generality. The computer vision methods used in a user interface for flying are not necessarily applicable to for example a user interface for platform jumping.

5.1 Computer Hearing

The computer hearing is the simplest part of the technology since we only need to detect whether the user is shouting or not. This can be done by simple thresholding. The first step is to compute the signal power in each 50ms window of audio samples delivered by the sound card's capture driver:

$$P(m) = \sum_{i=1}^{N} x(n)^2 , \qquad (1)$$

where *m* is the window index, *N* is the window length in samples and *x(n)* is the input signal. Since we are only interested in the signal power relative to the background noise power, we filter the power values using a first order IIR highpass filter derived from [13]

$$P_{hp}(m) = \frac{1}{1+\omega} P(m) - \frac{1}{1+\omega} P(m-1) + \frac{1-\omega}{1+\omega} P_{hp}(m-1) , \qquad (2)$$

where ω is the cutoff frequency of the filter in the range 0...1. With a constant level background noise $P_{hp}(m)$ is close to zero. When the user begins to shout, $P_{hp}(m)$ peaks and begins to decay exponentially. A threshold value *T* is used so that the avatar breathes fire as long as $P_{hp}(m)>T$.

The system detects all loud sounds as shouting, but it does not matter because falsely detected shouting has no negative consequences in the game. This is in agreement with the assumption 4 presented in chapter 3. The main benefit of the design is computational simplicity. Children also have fun experimenting with different sounds. For example in one test with three children, a boy found out that he could make QuiQui breath fire by clapping his hands. After that, the other children tried it out enthusiastically. According to our tests, especially young boys are excited about the voice interface.

Although the voice input mechanism is simple, it seems to increase the excitement of the game and supports the fantasy of being a dragon. Because of the direct mapping between shouting and breathing fire, the voice interface does alienate the user from the game. The interface allows the user to express himself or herself *as* the avatar, in contrast to voice interfaces that for example recognize commands spoken *to* the game.

5.2 Game Physics

The first step in the design of the motion analysis was to design game physics for the avatar. The physics determine the control signals computed from the user's movements. Our solution uses loosely interpreted Newtonian mechanics simulated with finite differences, that is, by replacing time derivatives of variables with differences between the variables at successive simulation steps. Note that when designing game physics, the real laws of physics are merely exemplary conceptual tools.

Our solution has two masses connected by a weightless rod as shown in Fig. 8. Instead of the center of mass the moments caused by the forces *F* and *G* are computed relative to the origin. The forces *G* denote gravity and *F* denotes the "thrust" force caused by the user's movements. The angle between the vector *F* and the rod equals the angle between the user's spine and the vertical axis. This allows the user to control the rotation of the avatar by bending sideways when flying. The mass at the lower end of the avatar is further away from the origin so that it causes the avatar to rotate back to upright position if the user is not moving.

Note that the avatar is not flexible like the user. The animation simply has three frames with hands held up, down and straight to left and right.

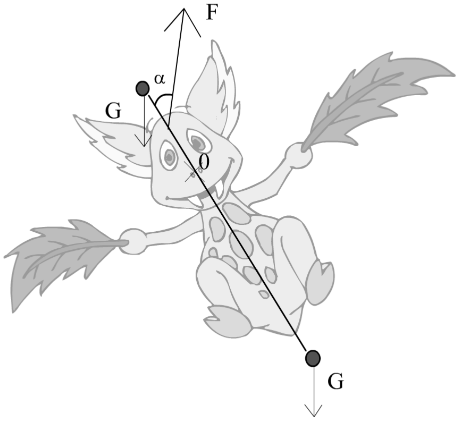

Fig. 8. Physics of the flying avatar

5.3 Computer Vision

To make the system robust to changes in the environment, the basic approach of the computer vision is differential motion analysis [20], based on intensity differences between consecutive video frames (temporal difference). The intensity differences are strongest at areas containing motion. Changes in lighting etc. cause only temporal disturbances in the system, opposite to for example the games by Intel and Mattel [4].

The drawback of using only the temporal difference data is that only the moving parts of the user can be detected. Fortunately, the analysis is simplified by the assumption that the movements of the user are limited to those relevant to the game context. Our system assumes that if the user is not simply standing still or walking around, he or she is trying to fly.

The system consists of the following processing stages, operating on 8-bit grayscale images:

1. Reduction of resolution to 120x96 pixels with spatial averaging (lowpass filtering) to reduce noise. The intensity of each output pixel is the average of an 8x8 pixel area of the input. The video is originally captured at 320x240 pixels or the closest resolution to that supported by the camera drivers.
2. Temporal differencing. Each output pixel is the absolute value of the difference of the corresponding input pixels of the last two input frames.
3. Thresholding. A threshold value is determined and pixel values below it are considered as noise and set to zero.
4. The image and the trajectory formed by its topmost nonzero pixels are analyzed to determine the output variables used to update the game physics.

Fig. 9 visualizes the operation, starting with the grayscale input image and ending with an image output by the third stage, augmented with visualization of the analysis of the fourth stage. The pose of the upper body is visualized as a straight line and the topmost nonzero pixels are plotted with maximum intensity. Note that the brightness of the images has been increased for better printing quality.

The threshold value of stage 3 is determined using histogram-based thresholding, various approaches of which are presented in [20]. It is assumed that the center of the background noise distribution is at the maximum of the histogram and that the distribution decreases monotonically at both sides of the maximum. The threshold is determined as the first local minimum of the histogram when searching from the maximum towards greater intensity values. According to our tests in various environments with a total of seven different camera models these assumptions seem to hold.

The first step in stage 4 is to detect whether the user is doing something or standing still. The criterion used is the amount of nonzero pixels. Analysis is continued only if the amount exceeds an experimentally determined value, currently the half of the image width.

The next step is to detect that the user is trying to fly instead of simply walking around. A parabola is fitted to the topmost pixels of the image in least-squares sense, as shown in Fig. 10. When the user is walking, the parabola usually fits well and its peak is located near the head of the user. The analysis is not continued if the quadratic coefficient of the parabola is negative enough and the peak of the parabola is horizontally near the center of the detected topmost pixels.

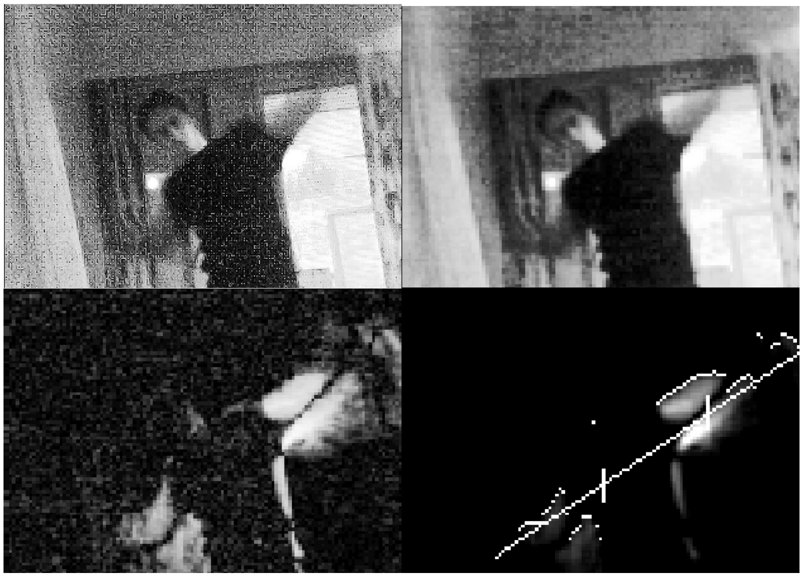

Fig. 9. The original input and the four processing states of the computer vision (from left to right and top to bottom)

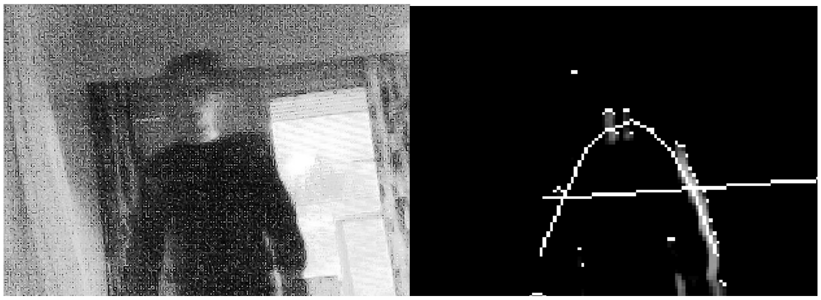

Fig. 10. Detecting a user walking

The analysis of the angle α and force F in Fig. 8 is based on image moments. Image moments are statistical characteristics that have been used for computer vision based game user interfaces by Freeman and Sengupta et al. [7][8][19]. An image moment of order *(p+q)* is computed as the summation [20]

$$m_{pq} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} x^p y^q i(x, y), \qquad (3)$$

where *i(x,y)* is the pixel intensity at coordinates *x* and *y*.

For user interface purposes, moments up to second order can be computed from silhouette representations of the user to obtain estimates of the size, location and orientation of the silhouette [7][8]. Freeman et al. also suggest that they could be computed from the temporal difference [8]. However, in this case the orientation information obtained from the moments is ambiguous, depending on the parts of the user that are moving. The principal axis of the difference may appear to be in the direction of the user's hands as well as the user's body.

Our solution for determining the pose of the upper body is to compute two spatially biased mass centers from the thresholded temporal difference. We use a left and right mass center, (x_{lett},y_{lett}) and (x_{right},y_{right}), marked with short vertical lines in the bottom right image in Fig. 9. The orientation of the user's shoulders is approximated with a line drawn through the centers, also shown in the figure.

The coordinates of the mass centers are computed from moments up to third order, by weighing the pixels differently depending on their position:

$$\begin{aligned} x_{right} &= \frac{1}{k_{right}} \sum_{x=1}^{w} \sum_{y=1}^{h} x(x^2 + y) i(x,y), \\ y_{right} &= \frac{1}{k_{right}} \sum_{x=1}^{w} \sum_{y=1}^{h} y(x^2 + y) i(x,y), \\ x_{left} &= \frac{1}{k_{left}} \sum_{x=1}^{w} \sum_{y=1}^{h} x((w-x)^2 + y) i(x,y), \\ y_{left} &= \frac{1}{k_{left}} \sum_{x=1}^{w} \sum_{y=1}^{h} y((w-x)^2 + y) i(x,y), \end{aligned} \tag{4}$$

where *h* and *w* are the height and width of the image in pixels. The scaling factors *k* equal the sum of the weights,

$$\begin{aligned} k_{right} &= \sum_{x=1}^{w} \sum_{y=1}^{h} (x^2 + y) i(x,y), \\ k_{left} &= \sum_{x=1}^{w} \sum_{y=1}^{h} ((w-x)^2 + y) i(x,y). \end{aligned} \tag{5}$$

Pixels closer to the left edge of the image are given more weight in the summation of the left mass center. The pixels closer to the top of the image are also given more weight in both mass centers to reduce the effect of the movement of the user's lower body. The *y* coordinates start from 1 at the bottom of the image, growing towards the top.

The position of the hands is practically impossible to detect accurately because of the motion blur. If the user's hands are moving fast enough, they can be perceived as an almost constant area of movement in the temporal difference. If flying is detected, the hand position information is changed by one animation frame according to the vertical movement of the mass center of the temporal difference. According to our tests this works sufficiently well.

Because of the ambiguity of the hand position information, the thrust force *F* in Fig. 8 cannot be computed from the angular velocity of the user's hands, although this would probably be the most intuitive interpretation. Instead, it is directly proportional to the amount of movement, measured as the amount of non-zero pixels. This is clearly a weakness, because it makes the system variant to changes in camera optics and the distance of the user. However, considering a single setup of the game, these properties vary only little. The user can adjust the overall sensitivity of the game in the main menu, which would be a useful feature even if the force could be determined more accurately. Because of the manual adjusting of the sensitivity, the requirement 1 presented in chapter 3 is only partially fulfilled. The game works with the default sensitivity, but in some cases it may feel too light or heavy.

The requirements 2, 3, 4 and 6 are met because of the use of temporal differencing and low-level image features. The system analyzes all movement, regardless of shape, color etc. The latency of the system is only one video frame.

Considering the requirement 5, the presence of other people than the user does not affect the game as long as they move less than the user so that the user dominates the analysis. However, the game is sensitive to other movement when the user wants to fall down and just stands still.

The system also supports collaborative play because the motion analysis based on image moments is vague enough to treat several users as one. However, this requires that the users are standing close enough to each other and flying in the same direction.

Considering the requirement 7, the computer vision is computationally efficient, leaving most of the processing power of current PC computers for the actual game. The implementation only caused an increase of 10% in the CPU load in our test system, compared to a bypassed situation, where a video stream was captured and converted to grayscale, but it was not fed through the processing stages described in this paper. The test system used was a 1GHz Pentium III Dell Inspiron 8100 laptop computer capturing video at 30fps using a Creative Video Blaster Webcam 5 camera.

To summarize, the computer vision solution is tested to work and enable an enjoyable game experience, although it is not perfect according to the requirements defined in chapter 3. The main drawback of the solution is that it is highly specific to the flying game.

6 Conclusions and Future Work

From the point of view of usability, we have defined a set of technical requirements for webcam and microphone based games for children in uncontrolled real-life environments like homes and daycare centers. We have also defined a set of simplifying assumptions to meet the requirements. An example game and user interface has then been designed and implemented based on the requirements, assumptions and testing with end users.

Iterative usability testing was carried out during the design. According to observations about the actions of the users, the user interface was redesigned to be more intuitive. The tests also showed that the child users act in a spontaneous and unpredictable way, which caused us to adapt a technically simplistic approach based on low-level image and audio features. The usability tests also pointed out general problems related to using camera based user interfaces in a space where the user may get out of the camera's view.

We are currently building more game and user interface prototypes and designing new computer vision algorithms, focusing on improving the collaborative aspects and robustness. The spatial design of perceptual user interfaces is also a topic that will be researched further.

The flying game described in this paper can be downloaded as an automatic installer package from the project's homepage *http://www.kukakumma.net*.

7 Acknowledgements

The work has been supported by YLE (Finland's national public service broadcasting company), Alfred Kordelin Foundation and University of Arts and Design Helsinki UIAH. The authors are currently receiving funding from the HeCSE and UCIT graduate schools of the Academy of Finland.

References

1. Blake, A., Isard, M., Active Contours The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion, Springer-Verlag London Limited, 1998
2. Bradski, G.R., Computer Vision Face Tracking For Use in a Perceptual User Interface, *Intel Technology Journal* Q2 '98. Available electronically at http://developer.intel.com
3. Crowley, J.L., Coutaz, J., Bérard, F., Things That See, *Communications of the ACM*, March 2000/Vol. 43, No. 3
4. D'Hooge, H., Game Design Principles for the Intel® Play™ Me2Cam* Virtual Game System, *Intel Technology Journal* Q4, 2001
5. Druin, A., Cooperative inquiry: Developing new technologies for children with children**.** In *Proceedings of CHI'99*, (1999) ACM/SIGCHI, N.Y., pp. 592-599
6. Druin, A., Children as our technology design partners: The surprising and the not-so-surprising, *Proceedings of International Workshop on Interaction Design and Children*, Aug. 28-29, 2002, Eindhoven, The Netherlands
7. Freeman, W.T., Tanaka, K., Ohta, J., Kyuma, K., Computer vision for computer games, *IEEE 2nd Intl. Conf. on Automatic Face and Gesture Recognition,* Killington, VT, October 1996
8. Freeman, W.T., Anderson, D., Beardsley, P., Dodge, C., Kage, H., Kyuma, K., Miyake, Y., Roth, M., Tanaka, K., Weissman, C., Yerazunis, W., Computer Vision for Interactive

Computer Graphics, *IEEE Computer Graphics and Applications,* May-June, 1998, pp. 42-53

9. Freeman, W.T., Beardsley, P.A., Kage, H., Tanake, K., Kyuma, K., Weissman, C.D., Computer Vision for Computer Interaction, *SIGGRAPH Computer Graphics Newsletter - Applications of Computer Vision to Computer Graphics,* Vol.33 No. 4 November 1999, ACM SIGGRAPH.
10. Gerig, J., Summary: Usability issues in computer games, http://www.listserv.acm.org/archives/wa.cgi?A2=ind0205E&L=chi-web&D=0&m=10716&P=720, 4th September 2002
11. Hanna, L., Risden, K., Alexander, K., Guidelines for Usability Testing with Children, *Interactions,* Vol.4 No.5 1997 pp. 9-14
12. Höysniemi, J., Hämäläinen, P., Turkki, L., Using Peer Tutoring in Evaluating the Usability of a Physically Interactive Computer Game with Children, *Proceedings of International Workshop on Interaction Design and Children*, Aug. 28-29, 2002, Eindhoven, The Netherlands
13. Ifeachor, E.C., Jervis, B.W., Digital Signal Processing A Practical Approach, Addison-Wesley, 1993, p. 400
14. Inkpen, K., Ho-Ching, W., Kuederle, O., Scott, S., Shoemaker, G., "This is fun! We're all best friends and we're all playing.": Supporting children's synchronous collaboration. Proceedings of *Computer Supported Collaborative Learning (CSCL) '99*. December 1999. Stanford, CA.
15. Krueger, M., Gionfriddo, T., Hinrichsen, K., VIDEOPLACE - An Artificial Reality, *Proceedings of CHI 85,* (1985) ACM/SIGCHI, N.Y. pp. 35-40
16. Moeslund, Thomas B.Computer vision-based human motion capture – a survey. Aalborg: Aalborg University, Laboratory of Computer Vision and Media Technology, 1999. (LIA Report; LIA 99-02.-ISSN 0906-6233)
17. Pinhanez, C.S., Wilson, A.D., Davis, J.W., Bobick, A.F., Intille, S., Blumberg, B., Johnson, M.P., Physically Interactive Story Environments, *IBM Systems Journal*, Vol. 39, Nos 3-4, 2000
18. Sinha, A.K., Landay, J.A., Visually Prototyping Perceptual User Interfaces through Multimodal Storyboarding, in *Proceedings of Workshop on Perceptual User Interfaces,* Nov. 15-16, 2001, Orlando, Florida
19. Sengupta, K., Wong, H. and Kumar, P. Computer Vision Games Using A Cheap (<100$) Webcam, in proceedings of *6th International Conference on Control, Automation, Robotics and Vision (ICARCV'2000)*, 5-8 December 2000, Singapore.
20. Sonka, M., Hlavac, V., Boyle, R., Image Processing, Analysis and Machine Vision, 2nd edition, Brooks/Cole Publishing Company, 1999
21. Stewart, J., Bederson, B.B, and Druin, A. (1999). Single display groupware: A model for co-present collaboration. *Proceedings of CHI 99.*(1999) ACM/SIGCHI, N.Y., pp. 286-293.
22. Subrahmanyam, K., Kraut, R.E., Greenfield, P.M, Gross, E.F., (2000), The impact of home computer use on children activities and development, *Children and computer technology*, vol. 10, No. 2, Fall/winter 2000.
23. Wren, C.R., Spacarino, F., Azarbayejani, A., Darrel, T., Davis, J., Starner, Kotani, A.,Chao, C., Hlavac, M., Russel, K., Bobick, A., Pentland, A., Perceptive Spaces for Performance and Entertainment (Revised), *ATR Workshop on Virtual Environments,* April 1998, Kyoto, Japan.
24. http://www.vividgroup.com, 4th May 2002.
25. http://www.interplay.com/games/product.asp?GameID=114, 5th September 2002
26. http://www.lieroextreme.com, 5th September 2002
27. http://www.tombraider.com, 5th September 2002